IN THE CLAIMS:

Presented below is a complete list of claims pending:

1. (Currently amended) A computer system comprising:

a processor having first and second execution cores and a check unit, the first and second execution cores to process instructions independently when the processor is in a split mode and to process identical instructions in lock step when the processor is in a redundant mode, and the check unit to compare results from the first and second execution cores when the processor is in redundant mode, the first execution core comprising a first instruction cache and the second execution core comprising a second instruction cache and a selector to select instructions from the first instruction cache in the redundant mode and to select instructions from the second instruction cache in the split mode; and

a non-volatile memory in which is stored a recovery routine that switches the processor to split mode when the check unit detects an error, identifies uncorrupted processor state data from at least one of the execution cores, and initializes the first and second execution cores with the identified processor state data.

2. (Original) The computer system of claim 1, wherein the error recovery routine returns the processor to the redundant mode.

3. (Original) The computer system of claim 1, wherein the error recovery routine identifies uncorrupted processor state data in selected storage structures associated with each execution

core, copies the uncorrupted processor state data to a specified memory location, and reconciles the copied processor state data, when the processor enters split mode.

4. (Currently amended) A method ~~for handling soft errors in a processor capable of operating first and second execution cores in redundant and split modes, the method~~ comprising:

in response to a core status bit, determining whether a processor having a first and a second execution cores is in a redundant mode or a split mode;

using a selector within the second execution core to select a first instruction from a first instruction cache within the first execution core if the processor is in the redundant mode; and

using the selector within the second execution core to select a second instruction from a second instruction cache within the second execution core if the processor is in the split mode.

~~detecting a soft error when the processor is operating in the redundant mode;~~

~~executing an error recovery routine on each execution core to save uncorrupted data from storage structures associated with the first and second execution cores, wherein the error recovery routine is stored in a non-volatile memory; and~~

~~recovering processor state data from the saved uncorrupted data.~~

5. (Currently amended) The method of claim [[4]] 14, further comprising:

reconciling the saved, uncorrupted data to recover the processor state data; and initializing the first and second execution cores using the recovered processor state data.

6. (Currently amended) A machine readable medium on which are stored instructions that are executable by a ~~dual execution core~~ processor having a first and a second execution cores to implement a method for recovering from soft errors, the method comprising:

in response to a core status bit, determining whether the processor is in the redundant mode or the split mode;

using a selector within the second execution core to select a first instruction from a first instruction cache within the first execution core if the processor is in the redundant mode;

using the selector within the second execution core to select a second instruction from a second instruction cache within the second execution core if the processor is in the split mode;

switching the processor to operate the ~~dual~~ first and second execution cores independently when a soft error is detected;

executing an error recovery routine on each of the first and second execution cores to recover a minimum set of processor state data from uncorrupted processor state data, wherein the error recovery routine is stored in a non-volatile memory; and

initializing each of the ~~dual~~ first and second execution cores, using the minimum set of processor state data.

7. (Original) The machine readable medium of claim 6, wherein initializing each of the dual execution cores comprises:

switching the processor to operate the dual execution cores in lock step; and

copying the minimum set of processor state data to each of the dual execution cores.

8. (Canceled).

9. (New) A processor comprising:

a first execution core operable in a split mode or a redundant mode, the first execution core comprising a first instruction cache; and

a second execution core to process instructions independent of the first execution core in the split mode and to process instructions in lock step with the first execution core in the redundant mode, the second execution core comprising a second instruction cache and a selector to select instructions from the first instruction cache in the redundant mode and to select instructions from the second instruction cache in the split mode.

10. (New) The processor of claim 9, further comprising:

a check unit to compare results from the first and second execution cores when the first and the second execution cores are in the redundant mode to detect errors.

11. (New) The processor of claim 10, further comprising:

a first set of one or more storage structures associated with the first execution core to store processor state data of the first execution core; and

a second set of one or more storage structures associated with the second execution core to store processor state data of the second execution core.

12. (New) The processor of claim 11, wherein the first and the second sets of one or more storage structures are parity-protected.

13. (New) The processor of claim 11, wherein, when the check unit detects an error, the first and the second execution cores switch to the split mode and save uncorrupted processor state data to an uncacheable portion of each of the first and second sets of one or more storage structures.

14. (New) The method of claim 4, further comprising:

detecting a soft error when the processor is operating in the redundant mode;

executing an error recovery routine on each execution core to save uncorrupted data from storage structures associated with the first and second execution cores, wherein the error recovery routine is stored in a non-volatile memory; and

recovering processor state data from the saved uncorrupted data.